IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Hansen, et al. Application No.: 10/757,925 Filed: January 16, 2004 For: METHOD AND SOFTWARE FOR PARTITIONED GROUP ELEMENT SELECTION OPERATION	Examiner: Jesse R. Moll Technology Center/Art Unit: 2181 Confirmation No.: 5116

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SECOND DECLARATION OF CRAIG HANSEN UNDER 37 CFR § 1.131

Sir:

I, Craig Hansen, hereby declare the following to be true:

BACKGROUND

1. I am the same Craig Hansen who submitted a Declaration of Craig Hansen Under 37 CFR § 1.131 dated July 31, 2009 (Hansen I declaration).

2. I have reviewed the Office Action dated March 19, 2009, in particular paragraph 4 thereof which states:

> Although the G.Select.8 instruction is shown to rearrange data based on a 64-bit selector, there is no proof that the rearrangement is done with "a plurality of fields each independently selecting one of the plurality of data elements" as recited in all independent claims.

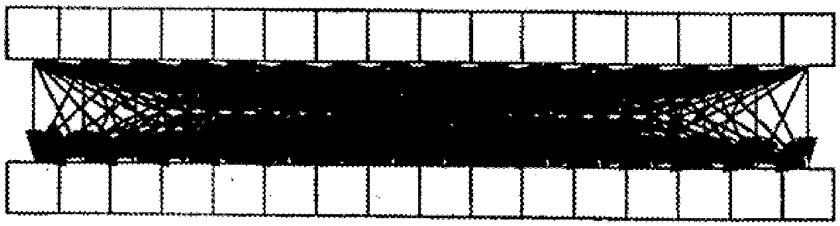
3. On page MU20396 of Exhibit 1 of my Hansen I declaration, the G.SELECT.8 instruction is disclosed as follows:

- G.SELECT.8

128 bits data, 4x16=64 bits control

16-way mux, byte-level granularity: complete byte permute

16-mux

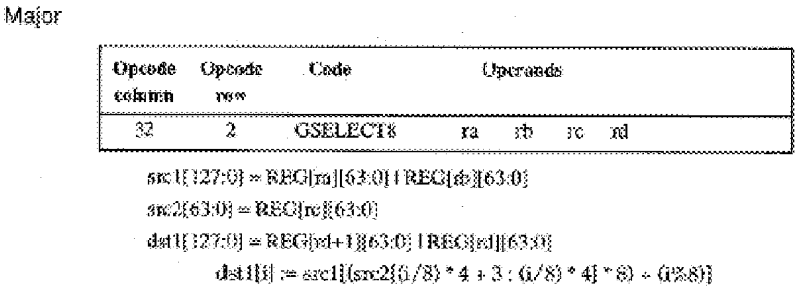


4. In this disclosure, the expression "4x16=64 bits control" describes the 64-bit selector, and indicates that it is comprised of 16 fields of 4 control bits each. The next line indicates that the operation being controlled is performed by a 16-way mux, which is illustrated, and that it has "byte-level granularity: complete byte permute." Since a byte is 8 bits of data, and the disclosure indicates that there are 128 bits of data total, it is clear from the disclosure that the data in question consists of 16 bytes. The fact that it is a "complete byte permute" indicates that each control field of 4 bits controls the destination of a byte of data. The figure displays a 16-way mux with a connection from each source to each destination, confirming that the assignments are independent. One skilled in the art would recognize this disclosure as a full disclosure of the elements listed in the Office Action as unproven, namely, that the rearrangement is done with a plurality of fields each independently selecting one of the plurality of data elements.

5. Exhibit A hereto is a draft version of the Euterpe MicroArchitecture Manual, which was reviewed by me and periodically updated prior to the filing of the present

application. Certain dates of Exhibit A showing conception have been redacted in accordance with standard practice and indicated with the phrase "REDACTED". Notwithstanding these redactions in Exhibit A, all of the redacted dates are prior to August 1, 1995.

6. The GSELECT8 instruction is disclosed on p. 23 of Exhibit A as follows:



7. Exhibit A uses pseudocode notation similar to the well-known C language (except that unlike C, the | symbol represents catenation of fields, := represents assignment of a value as in Pascal, and the notation x[y:z] represents a field comprised of bits y to z of x). As indicated in the pseudocode in Exhibit A, src1 (source register 1) provides 128 bits of source data from registers a and b (ra and rb), with the contents of register a catenated to the left of the contents of register b. Src2 (source register 2) provides the 64 bit control from register c (rc). Dst1 (destination register 1, which is placed into a register pair (rd+1,rd) specified by rd) receives 128 bits of data, which is specified for each of the bits 127 to 0. Bit "i" of Dst1 is selected from the source register under the control of the control register rc (src2), as defined by the formula given. In this formula, each bit "i" in the destination register is defined as the following bit in the source 1 register: (src2[(i/8)*4+3:(i/8)*4]*8)+(i%8). and shows selection of a bit field of four bits from src2 at bit field location ((i/8)*4+3:(i/8)*4), which is defined as a single value for each

group of 8 bits of i. (i/8) is the value i divided by 8 with the remainder discarded. Thus, for values of i from 0 to 7, (i/8) is zero, and the bit field used from src2 is src2[3:0], i.e. the low-order four bits. For values of i from 8 to 15, (i/8) is one, and the bit field used from src2 is src2[7:4], and so on, 'til for values of i from 120 to 127, (i/8) is fifteen, and the bit field used from src2 is src2[63:60]. It is apparent that the value of src2[3:0] is distinct from and independent of the value of src2[7:4], and so on to src2[63:60]; each of the four bit fields are independent. The contents of the selected field of src2 (a four bit field that has a value from zero to fifteen) is multiplied by 8, and (i%8), (the remainder when i is divided by 8) is added to this value to define which bit of src1 is extracted. For values of i from 0 to 7, and similarly for values of i from 8 to 15, and so on up to values from 120 to 127, (i%8) takes on values from 0 to 7, thus causing eight consecutive bits (an 8-bit byte) to be selected from src1 for each eight consecutive bits of dst1. It should also be apparent that the remainder of the expression: (src2[(i/8)*4+3:(i/8)*4]*8) does not change for each of the eight consecutive values of i from 0 to 7, then takes on a new value that does not change for each of the eight consective values of i from 8 to 15, and so on to the values of i from 120 to 127. In this fashion, the location of each 8-bit byte in the destination register is determined by a 4-bit control field, with control bits 0-3 selecting an 8-bit byte from any location in the source register and placing it in bits 0-7 of the destination register, control bits 4-7 selecting an 8-bit byte from any location in the source register, including the location previously selected, and placing it in bits 8-15 of the destination register, and so on until 128 bits consisting of 16 independently selected bytes have been selected from the source register and placed in the destination register as determined by the 4-bit control bits corresponding to each destination byte location.

8. It is clear from the pseudocode for the GSELECT8 instruction depicted in Exhibit A that the instruction rearranges data based on a 64-bit selector (src1), and more particularly, that the rearrangement is controlled by a plurality of 4-bit fields from the 64 selector bits in src2, and that each selector bit field in src2 independently selects one of the 8-bit bytes from src1 and places it in the corresponding byte location in dst1. The first 4-bit selector field selects the byte to be placed in the first 8 bits of the destination register, the second 4-bit selector field selects the byte to be placed in the second 8 bits of the destination register, and so on.

9. The phrase discussed above and quoted in paragraph 4 of the Office Action is clearly and fully disclosed to one skilled in the art in Exhibit 1 of the Hansen I declaration and Exhibit A of this declaration.

10. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and the these statements are made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, and any patent issuing thereon, or any patent to which this declaration is directed.

17-Sept-2009
Date

Craig Hansen